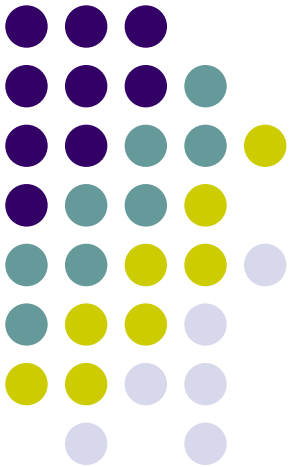
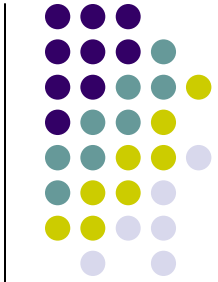
CSCI 2570 Introduction to Nanocomputing

Information Theory

John E Savage

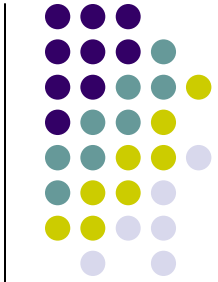


What is Information Theory



- Introduced by Claude Shannon. See Wikipedia
- Two foci: a) data compression and b) reliable communication through noisy channels.
- Data compression important today for storage and transmission, e.g., audio & video (JPEG).
- Reliable communication used in memories, CDs, Internet, and deep-space probes.

Source Models



- **Memoryless sources** generate successive outcomes that are independent and identically distributed.

- Source (S, **p**) has outcomes S = {1,2, …, n} that occur with probabilities **p** = $\{p_1, p_2, \ldots, p_n\}$

- E.g. **Binary Source:** S = {H,T}, $p_H = 1-p_T$

Entropy – A Measure of Information

- Entropy of source (S,**p**) in bits is

$$H(S) = -\sum_i p_i \log_2 p_i$$

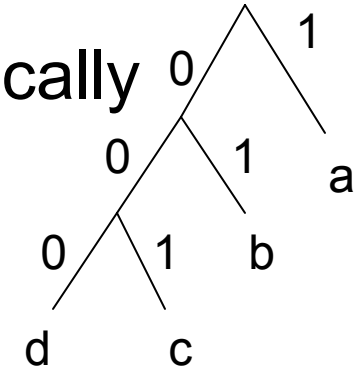
- **Binary Source:** H(S) = - p_H log p_H –(1- p_H)log (1- p_H)

- The larger the entropy, the less predictable is the source output and the more information is produced by seeing it.

- If base two logarithms used, entropy measured in bits (**bi**nary digi**ts**).

What are Codes?

- A **code** is a set of words (**codewords**).
- **Source codes** compress data probabilistically
  - E.g. Outputs, probabilities and codes:
    - P(a) =.5, P(b) = .25, P(c) = .125, P(d) =.125
    - w(a) =1, w(b) = 01, w(c) = 001, w(d) = 000



- **Channel codes** add redundancy for error correction and detection purposes.
  - 0 $\rightarrow$ 000, 1 $\rightarrow$ 111; decide by majority rule
  - Codes can be used for detection or correction.

Source Coding

- **Prefix condition:** no codeword is a prefix for another.
  - Needed to decode a source code.
- A source with *n* i.i.d. outputs and entropy H(S) can be compressed to a string of length *n*H(S) for large *n*.
- **Huffman coding algorithm** gives the most efficient prefix source encoding.
  - For *binary source code*, combine two least probable outcomes and give them both the same prefix.
  - Repeat using the prefix as a new outcome with probability equal to the sum of the two least probable outcomes.
  - Algorithm was used on previous page.

Source Coding Theorem

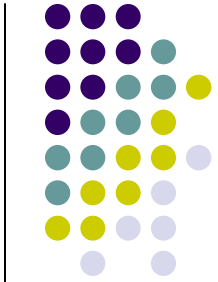
- **Theorem** Let codeword a_i over alphabet of b symbols encode ith output where $s_i = |a_i|$. Let $p(a_i)$ be probability of a_i. Let $E(X) = \sum_i s_i\, p(a_i)$ be average codeword length. Let

$$H(X) = -\sum_i p(a_i) \log p(a_i)$$

be the source entropy. Then,

$$\frac{H(X)}{\log b} \leq E(X) \leq \frac{H(X)}{\log b} + 1$$

Source Coding Theorem



Let $q_i = b^{-s_i}/C$ where C such that $\sum_i q_i = 1$

Using log x ≤ x-1, we have

$$\sum_i p_i \log(q_i/p_i) \leq 0$$

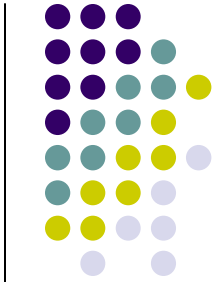
This implies

$$-\sum_i p_i \log p_i \leq -\sum_i p_i \log q_i$$

Using $q_i = b^{-s_i}/C$ we have

$$H(X) \leq E(X) \log b$$

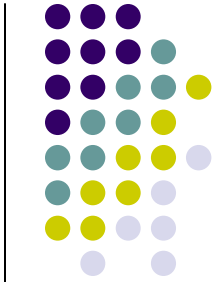
Source Coding Theorem



- Let codewords $\{a_i\}$, $s_i=|a_i|$, satisfy the prefix condition.

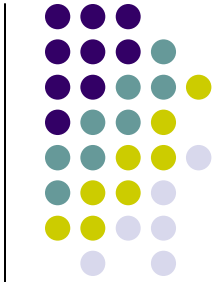
  Theorem Lengths $\{s_i\}$ satisfy Kraft's Inequality $\sum_i b^{-s_i} \leq 1$ and for any $\{s_i\}$ satisfying Kraft's Inequaliy, a prefix code can be constructed for them.

Source Coding Theorem



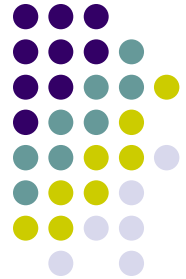
Proof Consider complete tree on *b* letters of depth $s_n = max_i\ s_i$. If A_i are leaves of the complete tree that are leaves of a_i, $|A_i| = b^{s_n - s_i}$ Since the number of descendants in the complete tree is exactly b^{s_n} and the A_i are disjoint, Kraft's Inequality follows.

Source Coding Theorem

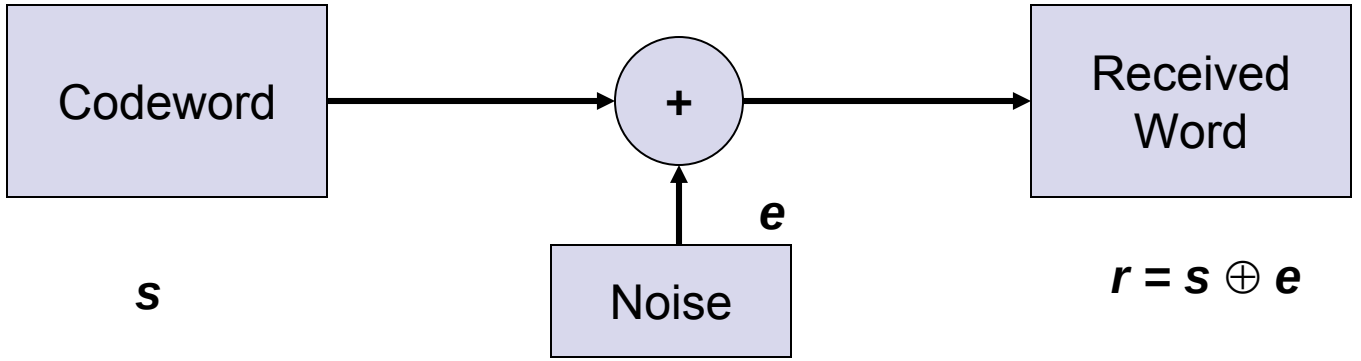


Proof(cont.) Let $s_1 \leq s_2 \leq \ldots \leq s_n$. To construct a prefix code, assign codeword to *n*th word, *w(n)*, which is the set of labels to a vertex at depth s_n. Assign a codeword to the *(n-1)*st word, *w(n-1)*, by picking a vertex at depth s_{n-1} and deleting all of its leaves in the complete tree. Continue in this fashion. The fact that Kraft's Inequality is satisfied ensures that this process can go to completion.

Discrete Memoryless Channels

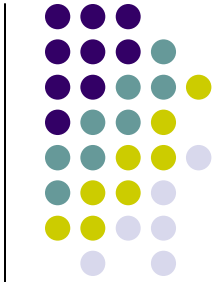


- Inputs are discrete; noise on successive transmissions is i.i.d.



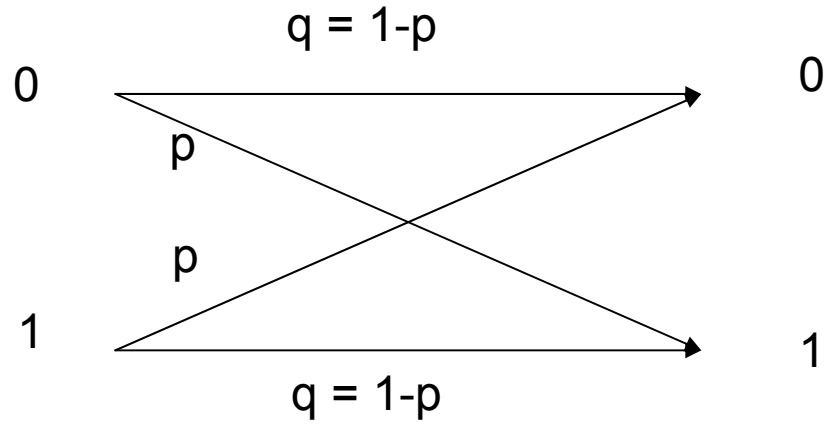
- Memoryless channels have a capacity, C, a maximum rate at which a source can transmit reliably through the channel, as we shall see.

Codes



- A **code** is a set of words (**codewords**).
  - Block codes and convolutional codes
- $(n,k,d)_q$ block codes
  - k inputs over alphabet of size q are encoded into codewords of length n over the same alphabet. The minimum Hamming distance (no. differences) between two codewords is d.
  - k = m**essage length**
  - n = **block length**
  - $R = k/n$ = **rate**
  - d = **minimum distance**
  - q = **alphabet size**

Binary Symmetric Channel



- ***r*** **= *s*** $\oplus$***e***. Error vector ***e*** identifies errors.
- Average number of errors in *n* transmissions is *np*. Standard deviation is $\sigma = (npq)^{1/2}$.
- If codewords are more than $2(np + t\sigma)+1$ bits apart, very likely can decode correctly.

Sphere Packing Argument

- "Likely error vectors" form sphere around each codeword.
- If the spheres are disjoint, the probability of decoding the received word correctly will be high.

Memoryless Channel Coding Theorem

- There exists an infinite family of codes of rate R < C such that *n*th code achieves a prob. Of error P(E) satisfying where *E(R) > 0* for R < C

$$P(E) \leq e^{-nE(R)}$$

- All codes with rate R > C require P(E) > ε > 0.

- Capacity of BSC

  C = 1 – H(p) = 1 + p log p + (1-p) log (1-p).

The Hamming Code - Example

- Encode $\boldsymbol{b} = (b_0, b_1, b_2, b_3)$ as $\boldsymbol{b}G$ where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

  - G is the **generator matrix**.
- This is a $(7,4,3)_2$ code. Why is $d = 3$?
  - Compare $\boldsymbol{b}_1G$ and $\boldsymbol{b}_2G$ where $\boldsymbol{b}_1 \neq \boldsymbol{b}_2$.
  - Note that $\boldsymbol{b}_1G \oplus \boldsymbol{b}_2G$ (term-by-term XOR) is equivalent to $\boldsymbol{b}_3G$ where $\boldsymbol{b}_3 = \boldsymbol{b}_1 \oplus \boldsymbol{b}_2$.

Other Methods of Reliable Communication

- Automatic Repeat Request (ARQ)
  - The receiver checks to see if the received is a codeword.
  - If not, it requests retransmission of the message.
  - This method can detect *d-1* errors when an (*n,k,d*) block code is used.
  - Requires buffering of data, which may result in loss of data.